On the Number of Linear Functions Composing Deep Neural Network: Towards a Refined Definition of Neural Networks Complexity

Yuuki Takai
RIKEN AIP
yuuki.takai@riken.jp

Akiyoshi Sannai
RIKEN AIP
akiyoshi.sannai@riken.jp

Matthieu Cordonnier
École Normale
Supérieure Paris-Saclay
matthieu.cordonnier
@ens-paris-saclay.fr

Abstract

The classical approach to measure the expressive power of deep neural networks with piecewise linear activations is based on counting their maximum number of linear regions. This complexity measure is quite relevant to understand general properties of the expressivity of neural networks such as the benefit of depth over width. Nevertheless, it appears limited when it comes to comparing the expressivity of different network architectures. This lack becomes particularly prominent when considering permutation-invariant networks, due to the symmetrical redundancy among the linear regions. To tackle this, we propose a refined definition of piecewise linear function complexity: instead of counting the number of linear regions directly, we first introduce an equivalence relation among the linear functions composing a piecewise linear function and then count those linear functions relative to that equivalence relation. Our new complexity measure can clearly distinguish between the two aforementioned models, is consistent with the classical measure, and increases exponentially with depth.

1 Introduction

Deep neural networks with rectified linear units (ReLU) as an activation function have been remarkably successful in computer vision, speech recognition, and other domains (Alex et al., 2012), (Goodfellow et al., 2013), (Wan et al., 2013), (Silver et al., 2017). However, the theoretical understanding to support this experimental progress is still insufficient, thereby motivating several researchers to bridge this crucial gap.

A fundamental theoretical problem is *the expressivity of neural networks*: given an architecture configuration (depth, width, layer type, activation function), which class of functions can neural networks compute and with what level of performance? To evaluate the expressive power of a neural network, we, therefore, need to define *a measure of its complexity*. In the case where ReLU is the only considered activation function, a neural network represents a piecewise linear function, thus, a natural method to measure complexity is to count the number of linear regions. From this perspective, we can theoretically justify favoring depth over width. Moreover, this has also been widely established through experimentation (Pascanu et al., 2013), (Montúfar et al., 2014), (Telgarsky, 2016), (Arora et al., 2016), (Eldan and Shamir, 2016), (Yarotsky, 2017), (Serra et al., 2018), (Chatziafratis et al., 2019), (Chatziafratis et al., 2020).

Nevertheless, as we subsequently show, using the number of linear regions as a straightforward measure, does not adequately reflect the properties of the underlying function which the network represents in some case.

Concretely, we consider permutation-invariant functions and the model introduced by Zaheer et al. (2017). This model has been proven to be a universal approximator for the class of permutation-invariant continuous functions (Maron et al., 2019), (Zaheer et al., 2017). Since this model is permutation-invariant, its expressive power is strictly lower than that of the fully connected model. However, we point out that the maximal number of linear regions for both the models is asymptotically similar.

This highlights the fact that the straightforward relationship between *number of linear regions* and *expressive power* needs to be qualified, because it cannot distinguish between these two models clearly. Thus, we propose a new complexity measure that enables us to reliably make this distinction.

Our main contribution is to introduce such a measure (Def-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

inition 2) and to prove that the invariant model and the fully connected model actually have different values (Theorem 1, Theorem 3). To define our measure, we consider not the *number of linear regions* but the *number of linear functions on them*. Our measure counts them relative to a certain equivalence relation. This relation identifies linear functions (and their inherent linear region) that can be mapped from one to another through a certain Euclidean transformation, i.e., isometric affine transformation. As a more intrinsic example, we consider digital images. The complexity of digital images is usually estimated by the number of pixels and the number of colors. The number of pixels corresponds to the conventional measure of complexity, i.e., the number of linear regions $c^{\#}$, and the number of colors corresponds to our proposed measure of complexity, i.e., the number of linear functions $c^{\sim}$. From this point of view, our measure is a natural consequence. We remark that other possible measures of complexity such as using Betti numbers of the linear regions (Bianchini and Scarselli, 2014), trajectories in the input space (Raghu et al., 2017), or the volumes of the boundaries of linear regions (Hanin and Rolnick, 2019) have been proposed.

The low expressive power of permutation-invariant shallow networks is translated to the fewness of linear functions (≈"colors") composing them, due to permutation invariance. Indeed, we show that for permutation-invariant shallow models, the proposed measure of complexity is the same as the number of orbits of linear regions by permutation action and that this number is relatively small. Our demonstration relies on theory of hyperplane arrangement which is stable by group action studied in (Kamiya et al., 2012). In Section 4, we modified the argument of Montúfar et al. (2014) to prove the benefit of depth over width. Particularly, the complexity of the proposed method increases exponentially with depth for fully connected deep models as well as for permutation-invariant deep models introduced by Zaheer et al. (2017).

2 Preliminaries and background

A (feedforward) neural network of depth $L+1$ is a composition of layers of units which defines a function $F\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_{L+1}}$ of the form

$$F(\boldsymbol{x}) = f_{L+1} \circ g_L \circ f_L \circ \cdots \circ g_1 \circ f_1(\boldsymbol{x}), \quad (2.1)$$

where $f_l\colon \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ is an affine map and $g_l\colon \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ is a nonlinear activation function. Throughout this paper, as activation functions, we consider the rectifier linear units (ReLU), i.e., for $\boldsymbol{x} = (x_1, \dots, x_{n_l})^\top \in \mathbb{R}^{n_l}$,

$$\mathrm{ReLU}(\boldsymbol{x}) = (\max\{0, x_1\}, \dots, \max\{0, x_{n_l}\})^\top \in \mathbb{R}^{n_l}.$$

Let $\underline{n} = (n_0, n_1, \dots, n_{L+1})$ and K be a connected compact n_0-dimensional subset of $\mathbb{R}^{n_0}$. Then, we define $\mathcal{H}_K^{\mathrm{full}}(\underline{n}) = \mathcal{H}_K^{\mathrm{full}}(n_0, n_1, \dots, n_{L+1})$ as the set of the restriction to K of the neural networks of the form (2.1) with $g_l = \mathrm{ReLU}$ for any $l = 1, \dots, L$. We call such a network a ReLU neural network. The affine transformation f_l can be written as $f_l(\boldsymbol{x}) = W_l\boldsymbol{x} + \boldsymbol{c}_l$ with a weight matrix $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ and a bias vector $\boldsymbol{c}_l \in \mathbb{R}^{n_l}$. We call the feedforward neural network *shallow* (resp. *deep*) if $L = 1$ (resp. $L > 1$).

Because ReLU is a continuous piecewise linear function, a function realized by a ReLU neural network is also continuous and piecewise linear. We are interested in the structures of such piecewise linear functions. Any piecewise linear function is encoded as the set of pairs made of a linear region and a linear function on it. Here, for a connected compact m-dimensional subset K in $\mathbb{R}^m$ and a piecewise linear function $f\colon K \to \mathbb{R}^n$, a connected region $D \subset K$ is called *a linear region of* f if f is linear on D and for any connected region $D' \subset K$ satisfying $D \subsetneq D'$, f is not linear on D'. For a piecewise linear function f, $c^{\#}(f)$ denotes the number of linear regions of f. For a set of piecewise linear functions $\mathcal{H}$, we set $c^{\#}(\mathcal{H}) = \max\{c^{\#}(f) \mid f \in \mathcal{H}\}$.

2.1 The number of linear regions for shallow fully connected neural networks

To calculate the maximum number of linear regions for shallow ReLU neural networks, we use arguments from hyperplane arrangement theory as in (Pascanu et al., 2013). Let us consider a shallow ReLU neural network $F \in \mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2)$, i.e., a network of the form

$$F(\boldsymbol{x}) = f_2 \circ g_1 \circ f_1(\boldsymbol{x}), \quad (2.2)$$

where $f_1\colon K \to \mathbb{R}^{n_1}$ and $f_2\colon \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ are two affine maps and $g_1\colon \mathbb{R}^{n_1} \to \mathbb{R}^{n_1}$ is ReLU.

The linear regions of F depend only on the affine map f_1. We write $f_1(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{c}$ for $W = (a_{ij}) \in \mathbb{R}^{n_1 \times n_0}$ and $\boldsymbol{c} = (c_i) \in \mathbb{R}^{n_1}$. Let H_i be the hyperplane in $\mathbb{R}^{n_0}$ defined as

$$a_{i1}x_1 + \cdots + a_{in_0}x_{n_0} + c_i = 0 \ \cdots \ H_i$$

for $i = 1, \dots, n_1$. Then, the linear regions of F are exactly the chambers of the hyperplanes arrangement defined by $\mathcal{A} = \{H_1, \dots, H_{n_1}\}$, i.e., the connected components of the complement $\mathbb{R}^{n_0} \setminus \bigcup_i H_i$. Let $\mathrm{Ch}(\mathcal{A})$ denotes the set of chambers of arrangement $\mathcal{A}$. Then, Schläfli showed that the cardinality $|\mathrm{Ch}(\mathcal{A})|$ of $\mathrm{Ch}(\mathcal{A})$ satisfies

$$|\mathrm{Ch}(\mathcal{A})| \leq \sum_{i=0}^{n_0} \binom{n_1}{i} \quad (2.3)$$

and the equality holds if $\mathcal{A}$ is in general position (Orlik and Terao, 2013, Introduction). Here, we say that the hyperplane arrangement $\mathcal{A} = \{H_1, \dots, H_{n_1}\}$ is *in general position* if $\mathcal{A}$ satisfies that for any $r = 1, \dots, n_0$, the codimension of the intersection $H_{i_1} \cap \cdots \cap H_{i_r}$ is equal to r if $r \leq n_1$ and $H_{i_1} \cap \cdots \cap H_{i_r} = \emptyset$ if $r > n_1$ (see

Appendix A.1 for an illustration). For the hyperplane arrangement $\mathcal{A}$ defined by the fully connected shallow neural network above, we remark that it is always possible to make it being in general position by perturbing the weight matrix W and the bias vector $\boldsymbol{c}$. Moreover, for any connected compact n_0-dimensional subset $K \subset \mathbb{R}^{n_0}$ and a hyperplane arrangement $\mathcal{A}$, we can take another hyperplane arrangement $\mathcal{A}' = \{H_i' \mid i = 1, \dots, n_1\}$ such that $|\mathrm{Ch}(\mathcal{A})|$ is equal to the number of connected components of $K \setminus \bigcup_i (K \cap H_i')$ by translating or scaling $\mathcal{A}$ if it is necessary. In particular, the maximal number $c^{\#}(\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2))$ of linear regions of the fully connected shallow ReLU neural network having a n_0-dimensional input layer and a n_1-dimensional hidden layer is $\sum_{i=0}^{n_0} \binom{n_1}{i}$. For n_0 such that $0 \leq n_0 \leq n_1/2$, by (Ash, 1965, Section 4.7), the estimate of the sum of binomial coefficients is

$$\begin{aligned} \frac{2^{n_1 H(n_0/n_1)}}{\sqrt{8 n_0 (1 - n_0/n_1)}} \leq \binom{n_1}{n_0} \leq c^{\#}(\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2)) \\ = \sum_{i=0}^{n_0} \binom{n_1}{i} \leq 2^{n_1 H(n_0/n_1)}, \quad (2.4) \end{aligned}$$

where $H(p)$ is the binary entropy function defined as

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

for $0 < p < 1$ and $H(0) = H(1) = 0$.

2.2 The number of linear regions for the permutation invariant model

We review the permutation invariant shallow model introduced in (Zaheer et al., 2017) and show that this model can have as many linear regions as a fully connected shallow neural network, though this model has a lower expressive power than the fully connected model. We illustrate this calculation on a simple example in Appendix A.2.

We define the permutation action on $(\mathbb{R}^n)^m$ of permutation group S_n by the following way. For $\sigma \in S_n$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_m) \in (\mathbb{R}^n)^m$ where $\boldsymbol{x}_i = (x_{i1}, \dots, x_{in})^\top \in \mathbb{R}^n$, we define

$$\begin{aligned} \sigma \cdot \boldsymbol{X} &= (\sigma \cdot \boldsymbol{x}_1, \dots, \sigma \cdot \boldsymbol{x}_n), \\ \sigma \cdot \boldsymbol{x}_i &= (x_{i\sigma^{-1}(1)}, \dots, x_{i\sigma^{-1}(n)})^\top. \end{aligned}$$

For a subset K of $\mathbb{R}^n$, we say that K is stable by permutation action if for any $\boldsymbol{x} \in K$, $\sigma \cdot \boldsymbol{x} \in K$ holds for any $\sigma \in S_n$.

We consider a permutation invariant shallow network as

$$F(\boldsymbol{x}) = f_2 \circ g_1 \circ f_1(\boldsymbol{x}) \quad (2.5)$$

where $f_1 \colon \mathbb{R}^n \to (\mathbb{R}^n)^m$ is a permutation equivariant affine map, i.e., $f_1(\sigma \cdot \boldsymbol{x}) = \sigma \cdot f_1(\boldsymbol{x})$ for any $\sigma \in S_n$ and $\boldsymbol{x} \in \mathbb{R}^n$, and $f_2 \colon (\mathbb{R}^n)^m \to \mathbb{R}^{m'}$ is a permutation invariant affine map i.e., $f_2(\sigma \cdot \boldsymbol{X}) = f_2(\boldsymbol{X})$ for any $\boldsymbol{X} \in (\mathbb{R}^n)^m$ and $\sigma \in S_n$, and $g_1 \colon (\mathbb{R}^n)^m \to (\mathbb{R}^n)^m$ is ReLU. Then, the realized function F is permutation invariant, i.e., $F(\sigma \cdot \boldsymbol{x}) = F(\boldsymbol{x})$ for any $\sigma \in S_n$. Let K be a connected compact n-dimensional subset of $\mathbb{R}^n$ which is stable by permutation action. Then, we define $\mathcal{H}_K^{\mathrm{inv}}(n, mn, m')$ as the set of the restrictions to K of the permutation invariant ReLU neural networks of the form (2.5). By universal approximation theorem (Maron et al., 2019), any permutation-invariant, continuous function on K can be approximated by such a neural networks for large enough m'.

The set of linear regions of the model depends only on the affine map f_1 as in the fully connected case. Using (Zaheer et al., 2017, Lemma 3), by the permutation equivariance of f_1, if we set $f_1(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{c}$ for some $W \in (\mathbb{R}^{n \times n})^m$ and $\boldsymbol{c} \in (\mathbb{R}^n)^m$, these W and $\boldsymbol{c}$ can be written as

$$W = \begin{pmatrix} a_1 I + b_1 (I - \mathbf{1}\mathbf{1}^\top) \\ \vdots \\ a_m I + b_m (I - \mathbf{1}\mathbf{1}^\top) \end{pmatrix}, \ \boldsymbol{c} = \begin{pmatrix} c_1 \mathbf{1} \\ \vdots \\ c_m \mathbf{1} \end{pmatrix}$$

for some $a_1, \dots, a_m, b_1, \dots, b_m, c_1, \dots, c_m \in \mathbb{R}$. Here, I is the identity matrix in $\mathbb{R}^{n \times n}$ and $\mathbf{1}$ is the all one vector in $\mathbb{R}^n$. Thus, the set of linear regions of F is equal to the set of chambers of the hyperplanes arrangement $\mathcal{B}_{m,n} = \{H_{11}, \dots, H_{mn}\}$ defined for $i = 1, \dots, m$ by,

$$\begin{cases} a_i x_1 + b_i x_2 + \cdots + b_i x_n + c_i = 0 & \cdots H_{i1} \\ b_i x_1 + a_i x_2 + \cdots + b_i x_n + c_i = 0 & \cdots H_{i2} \\ \qquad\qquad \vdots & \\ b_i x_1 + b_i x_2 + \cdots + a_i x_n + c_i = 0 & \cdots H_{in}. \end{cases} \quad (2.6)$$

We calculate the number of chambers of the arrangement $\mathcal{B}_{m,n}$. As in inequation (2.3), the number of chambers are bounded from above by $\sum_{i=0}^{n} \binom{mn}{i}$ and attains this bound if the arrangement $\mathcal{B}_{m,n}$ is in general position. However, this arrangement $\mathcal{B}_{m,n}$ cannot be in general position. Indeed, the hyperplanes in the arrangement $\mathcal{B}_{m,n}$ satisfy

$$H_{i_1,j} \cap H_{i_2,j} \cap H_{i_3,j} = \emptyset, \quad (2.7)$$

$$\begin{aligned} H_{i_1,j_1} \cap H_{i_1,j_2} \cap H_{i_2,j_1} &= H_{i_1,j_1} \cap H_{i_1,j_2} \cap H_{i_2,j_2} \\ &= H_{i_1,j_1} \cap H_{i_2,j_1} \cap H_{i_2,j_2} \end{aligned} \quad (2.8)$$

for $i_1, i_2, i_3 = 1, \dots, m$ and $j, j_1, j_2 = 1, \dots, n$. Nevertheless, we can calculate the number of chambers of the arrangement $\mathcal{B}_{m,n}$ by applying the Deletion-Restriction theorem (Theorem 1 in Appendix B) (Orlik and Terao, 2013, Theorem 2.56 and Theorem 2.68) under the assumption (2.7) and (2.8). The detail of the calculation is in Appendix B.

Proposition 1. *We assume that $m > n/2$. Then, the maximum $b_{m,n}$ of the number of chambers of $\mathcal{B}_{m,n}$ is bounded from below by a function $g(m, n)$ which is polynomial with respect to m of degree n, and which the coefficient of the leading term is bounded from below by $(2^{5/4})^n/(n\sqrt{2})$.*

2.3 Comparison of the numbers of linear regions

To equalize the number of hidden units in both models, we consider the fully connected model (2.2) withd $n_0 = n$ and $n_1 = mn$. Let K be a connected compact n-dimensional subset of $\mathbb{R}^n$ which is stable by permutation action. By a universal approximation theorem (Sonoda and Murata, 2017), if we increase the number of hidden units of the fully connected shallow models, the elements of $\mathcal{H}_K^{\text{full}}(n, mn, m')$ can approximate any continuous maps on a compact set of $\mathbb{R}^n$. On the other hand, although elements of $\mathcal{H}_K^{\text{inv}}(n, mn, m')$ are also universal approximators for permutation-invariant functions (Maron et al., 2019), any function which is not permutation-invariant cannot be approximated by the elements of $\mathcal{H}_K^{\text{inv}}(n, mn, m')$. This implies that the expressive power of the permutation-invariant shallow models is strictly lower than for the fully connected shallow models.

In keeping with this observation, we compare maximum number of linear regions for the fully connected (2.2) and the permutation invariant shallow models (2.5). By the estimate (2.4) with $n_0 = n$ and $n_1 = mn$, we have

$$c^{\#}(\mathcal{H}_K^{\text{full}})(n, mn, n') \geq \frac{2^{mnH(1/m)}}{\sqrt{8n(1-1/m)}} \geq \frac{e^n}{2\sqrt{2n}} m^n + O(m^{n-1}).$$

On the other hand, by Proposition 1, the maximal number of linear regions of permutation invariant shallow models is also bounded from below as

$$c^{\#}(\mathcal{H}_K^{\text{inv}})(n, mn, n') \geq \frac{(2^{5/4})^n}{n\sqrt{2}} m^n + O(m^{n-1}).$$

In particular, although there is a difference of bases, $c^{\#}(\mathcal{H}_K^{\text{inv}})(n, mn, n')$ does also increase exponentially with respect to n. This means that the maximum numbers of linear regions cannot represent the difference of expressive powers of these models clearly.

This observation indicates that we should consider some refined measure for complexity and expressive power to be able to distinguish between these two classes of models more clearly.

3 Measure of complexity as the numbers of equivalent classes of linear functions

In this section, we introduce a measure of complexity which can distinguish permutation-invariant shallow models from fully connected shallow models. Before proposing a refined measure of complexity, we observe the structure of piecewise linear functions that are permutation-invariant.

Let K be a connected compact n-dimensional subset of $\mathbb{R}^n$ which is stable by permutation action and $f\colon K \to \mathbb{R}^{n'}$ be a piecewise linear function which is permutation invariant by the permutation group S_n, and $\mathcal{F}(f) = \{(f_\lambda, D_\lambda) \mid \lambda \in \Lambda\}$ the set of pairs of linear regions $D_\lambda \subset K$ of f and the linear associated function f_λ on D_λ, i.e., f_λ is the restriction $f|_{D_\lambda}$ of f on D_λ. We call this set $\mathcal{F}(f)$ *the set of linear functions of* f. We often abbreviate an element $(f_\lambda, D_\lambda) \in \mathcal{F}(f)$ to f_λ. Then, it is easy to show that for any permutation $\sigma \in S_n$ and any linear region D of f, the image $\sigma(D)$ of D by σ is also a linear region. By this fact and the permutation invariance of f, for any (f_λ, D_λ) and $\sigma \in S_n$, there is a λ' such that $\sigma(D_\lambda) = D_{\lambda'}$ and $f_\lambda = f_{\lambda'} \circ \sigma|_{D_\lambda}$. Here, we regard the permutation σ as a linear transformation on $\mathbb{R}^n$. Then, the linear transformation induced by permutation σ is isometric with respect to L^2-norm, because the map taking L^2-norm $\boldsymbol{x} \mapsto \|\boldsymbol{x}\|_2$ is permutation-invariant.

Inspired by this observation, we define an equivalence relation $\sim$ on the set of pairs $\mathcal{F}(f)$ of linear functions and regions for piecewise linear function $f\colon K \to \mathbb{R}^{n'}$ as follows:

Definition 1. *Let $f\colon K \to \mathbb{R}^{n'}$ be a piecewise linear function and $\mathcal{F}(f) = \{(f_\lambda, D_\lambda) \mid \lambda \in \Lambda\}$ the set of the linear functions of f. Then, we say that f_λ is equivalent to $f_{\lambda'}$, denoted by $f_\lambda \sim f_{\lambda'}$, if there is a Euclidean transformation $\phi\colon \mathbb{R}^n \to \mathbb{R}^n$ satisfying (1) $\phi(D_\lambda) = D_{\lambda'}$ and (2) $f_\lambda = f_{\lambda'} \circ \phi|_{D_\lambda}$. Here, a Euclidean transformation ϕ is an affine map written as $\phi(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ for an orthogonal matrix A and a vector $\boldsymbol{b}$.*

We can characterize the invariant function for a group action as follows: (the proof is in Appendix C):

Proposition 2. *Let f be a piecewise linear function on K and $\mathcal{F}(f) = \{(f_\lambda, D_\lambda) \mid \lambda \in \Lambda\}$ the set of linear functions of f. We assume that there is a set $\Phi = \{\phi_1, \dots, \phi_t\}$ of Euclidean transformations on $\mathbb{R}^n$ such that for any $\phi \in \Phi$ and any linear regions D_λ, there is a $\lambda' \in \Lambda$ such that (1) $\phi(D_\lambda) = D_{\lambda'}$, (2) $f_\lambda = f_{\lambda'} \circ \phi|_{D_\lambda}$. Then, f is $\hat{\Phi}$-invariant, where $\hat{\Phi} = \langle \phi_1, \dots, \phi_t \rangle$ is the group generated by Φ.*

The relation $\sim$ is an equivalence relation. Then, we propose the following measure of complexity:

Definition 2. *We define* the measure of complexity $c^{\sim}(f)$ of f *by the number of equivalent classes $\mathcal{F}(f)/\sim$. For a set $\mathcal{H}$ of piecewise linear functions, we define* the measure of complexity $c^{\sim}(\mathcal{H})$ of $\mathcal{H}$ *by the maximum of $c^{\sim}(f)$ for any $f \in \mathcal{H}$.*

As a trivial upper bound, $c^{\sim}(f)$ is bounded from above by $|\mathcal{F}(f)|$, i.e., the number of linear regions. More generally, the set $\mathcal{F}(f)$ of linear functions of f may be infinite. However, if f is realized by a ReLU neural network of finite width and finite depth, $\mathcal{F}(f)$ is finite.

We calculate this measure of complexity for the previous two classes of models. We remark that any Euclidean transformation ϕ does not change the volumes of linear regions.

Thus, if the volumes of two linear regions D_λ and $D_{\lambda'}$ are different, then f_λ and $f_{\lambda'}$ cannot be equivalent. We use this observation later to count the number of equivalent classes.

3.1 Examples of our measure of complexity in 1-dimensional case

Here, we show some examples in the 1-dimensional case and calculate our measures of complexity. For simplicity, we consider them on the interval $K = [0, 1]$.

Let f be a piecewise linear function on $[0, 1]$ and $[0, 1] = \bigcup_{i=1}^{m} D_i$ be the decomposition by linear regions D_i for f and $f_i = f|_{D_i}$. Then, $(f_i, D_i) \sim (f_j, D_j)$ holds only if $|D_i| = |D_j|$, where $|D|$ for interval $D = [p, q]$ is the length $q - p$. Indeed, by Definition 1, there is a Euclidean transformation $\phi\colon \mathbb{R}^1 \to \mathbb{R}^1; x \mapsto ax + b$ such that $\phi(D_j) = D_i$. As ϕ is Euclidean transformation, a is equal to ± 1. If we set $D_i = [p_i, q_i]$, then $\phi(D_j) = D_i$ is equivalent to $[p_j, q_j] = [\phi(p_i), \phi(q_i)] = [ap_i + b, aq_i + b]$. As $a = \pm 1$, this implies that $|D_j| = |D_i|$. Moreover, then, $ap_i + b = p_j$ holds. In particular, $b = p_j - ap_i$ holds. Because a is 1 or -1, there are only two choices of the Euclidean transformation $\phi\colon D_i \to D_j$.

Furthermore, we set $f_i(x) = \alpha_i x + \beta_i$ for $i = 1, \dots, m$. Then, $(f_j \circ \phi)|_{D_i} = f_i$ holds. Hence, for $x \in D_i$,

$$\alpha_i x + \beta_i = \alpha_j(ax + b) + \beta_j = a\alpha_j x + \alpha_j b + \beta_j$$

holds. Thus, we have $\alpha_i = a\alpha_j$ and $\beta_i = b\alpha_j + \beta_j$.

By combining these arguments, there are at most two linear functions on $D_j = [p_j, q_j]$ equivalent to (f_i, D_i) where $f_i(x) = \alpha_i x + \beta_i$ and $D_i = [p_i, q_i]$: For $f_i(x) = \alpha_i x + \beta_i$,

$$f_j(x) = \begin{cases} \alpha_i x + \beta_i - (p_j - p_i)\alpha_i & \text{if } a = 1, \\ -\alpha_i x + \beta_i + (p_j + p_i)\alpha_i & \text{if } a = -1. \end{cases}$$

Based on this observation, we show three examples on $[0, 1]$.

Example 1. *Let $f\colon [0, 1] \to \mathbb{R}$ be the piecewise linear function defined by $\{(f_i, D_i), i = 1, 2, 3, 4\}$, where $D_1 = [0, 1/4]$, $D_2 = [1/4, 1/2]$, $D_3 = [1/2, 3/4]$, $D_4 = [3/4, 1]$ and*

$$\begin{cases} f_1(x) = \alpha x + \beta & \text{for } x \in D_1, \\ f_2(x) = \alpha(x - 1/4) + \beta & \text{for } x \in D_2, \\ f_3(x) = \alpha(x - 1/2) + \beta & \text{for } x \in D_3, \\ f_4(x) = \alpha(x - 3/4) + \beta & \text{for } x \in D_4. \end{cases}$$

The function is drawn on Figure 1. By the Euclidean transformation $\phi\colon [0, 1/4] \to [1/4, 1/2]$, $x \mapsto x + 1/4$, $(f_1, D_1) \sim (f_2, D_2)$ holds. Similarly, $(f_i, D_i) \sim (f_1, D_1)$ holds for any i. Hence, $c^{\sim}(f) = 1$.

Example 2. *Let $f\colon [0, 1] \to \mathbb{R}$ be the piecewise linear function defined by $\{(f_i, D_i), i = 1, 2, 3, 4\}$, where $D_1 = [0, 1/4]$, $D_2 = [1/4, 1/2]$, $D_3 = [1/2, 3/4]$, $D_4 = [3/4, 1]$ and*

$$\begin{cases} f_1(x) = \alpha x + \beta & \text{for } x \in D_1, \\ f_2(x) = -\alpha(x - 1/4) + \alpha/4 + \beta & \text{for } x \in D_2, \\ f_3(x) = \alpha(x - 1/2) + \beta & \text{for } x \in D_3, \\ f_4(x) = -\alpha(x - 3/4) + \alpha/4 + \beta & \text{for } x \in D_4, \end{cases}$$

as Figure 2. By the Euclidean transformation $\phi\colon [0, 1/4] \to [1/4, 1/2]$, $x \mapsto -x + 1/2$, $(f_1, D_1) \sim (f_2, D_2)$ holds. Similarly, $(f_i, D_i) \sim (f_1, D_1)$ holds for any i. Hence, $c^{\sim}(f) = 1$.

Example 3. *Let $f\colon [0, 1] \to \mathbb{R}$ be the piecewise linear function defined by $\{(f_i, D_i), i = 1, 2, 3, 4\}$, where $D_1 = [0, 1/7]$, $D_2 = [1/7, 2/5]$, $D_3 = [2/5, 2/3]$, $D_4 = [2/3, 1]$ as Figure 3. Then, we have $|D_1| = 1/7$, $|D_2| = 9/35$, $|D_3| = 4/15$, and $|D_4| = 1/3$. By the above argument in the beginning of this section, there is no Euclidean transformation ϕ such that $\phi(D_j) = D_i$ for any $i \neq j$. Hence, $c^{\sim}(f) = 4$.*

Example 4. *Let $f\colon [0, 1] \to \mathbb{R}$ be the piecewise linear function defined by $\{(f_i, D_i), i = 1, 2\}$, where $D_1 = [0, 1/2]$, $D_2 = [1/2, 1]$,*

$$\begin{cases} f_1(x) = \alpha_1 x + \beta_1 & \text{for } x \in D_1, \\ f_2(x) = \alpha_2 x + \beta_2 & \text{for } x \in D_2, \end{cases}$$

such that $|\alpha_1| \neq |\alpha_2|$ as Figure 4. Then, there is no Euclidean transformation ϕ such that $\phi(D_1) = D_2$. Hence, $c^{\sim}(f) = 2$.

Example 5. *Let $f\colon [0, 1] \to \mathbb{R}$ be the piecewise linear function defined by $\{(f_i, D_i), i = 1, 2\}$, where $D_1 = [0, 1/2]$, $D_2 = [1/2, 1]$,*

$$\begin{cases} f_1(x) = \beta_1 & \text{for } x \in D_1, \\ f_2(x) = \beta_2 & \text{for } x \in D_2, \end{cases}$$

such that $\beta_1 \neq \beta_2$ as Figure 5. Then, there is no Euclidean transformation ϕ such that $\phi(D_1) = D_2$. Thus, $c^{\sim}(f) = 2$.

3.2 Fully connected shallow models

In this subsection, we show the existence of a fully connected model as (2.2) for which the proposed complexity is equal to $\sum_{i=0}^{n_0} \binom{n_1}{i}$. Let F be a ReLU shallow neural network model as (2.2) and $\mathcal{F}(F) = \{(F_i, D_i) \mid i = 1, \dots, N\}$ be the set of linear functions of F. As remarked above, by the condition of Definition 1 (1), if the volumes of two linear regions D_i and D_j are different, the corresponding linear functions F_i and F_j cannot be equivalent. Therefore, if all the linear regions D_i have different volumes, all the equivalence class of F are singletons, and its complexity $c^{\sim}(F)$ is equal to the number N of linear regions. By perturbing the weight matrix W or the bias vector $\boldsymbol{c}$, we can make F satisfy this condition. Hence, the

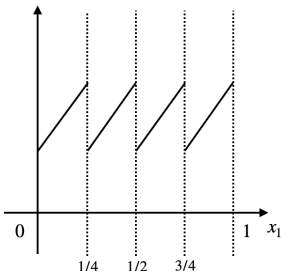

Figure 1: Example 1

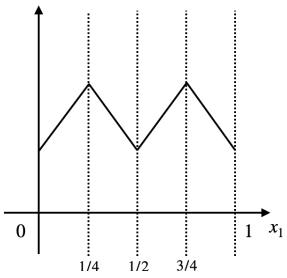

Figure 2: Example 2

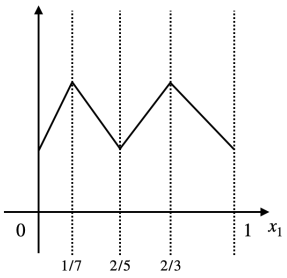

Figure 3: Example 3

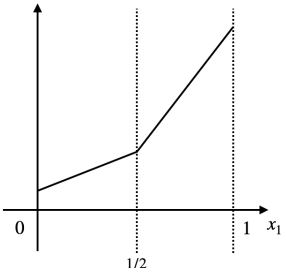

Figure 4: Example 4

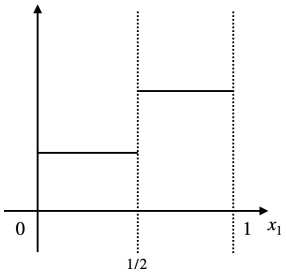

Figure 5: Example 5

measure of the complexity $c^{\sim}(\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2))$ of fully connected shallow ReLU neural networks remains equal to $\sum_{i=0}^{n_0} \binom{n_1}{i}$.

Theorem 1. *The measure of complexity of* $\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2)$ *is equal to* $c^{\#}(\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2))$. *In particular, the following holds:*

$$c^{\sim}(\mathcal{H}_K^{\mathrm{full}}(n_0, n_1, n_2)) \geq \frac{2^{n_0 H(n_1/n_0)}}{\sqrt{8n_0(1 - n_0/n_1)}}.$$

3.3 Permutation invariant models

Next, we consider this complexity for the permutation-invariant model. In this case, the permutation action of permutation group S_n induces equivalence on linear regions. This effect causes the gap between our complexity and the number of linear regions. In particular, for a permutation-invariant model F, the complexity $c^{\sim}(F)$ is equal to the number of orbits of linear regions via permutation action. To calculate the number of orbits of linear regions, we use arguments from *group action stable* hyperplanes arrangement theory investigated in (Kamiya et al., 2012). See Appendix A.3 for an illustration on a simple example.

Let K be a connected compact n-dimensional subset of $\mathbb{R}^n$ which is stable by permutation action. As in Section 2.2, the linear regions of the restriction to K of permutation-invariant model F defined in (2.5), are the chambers of the hyperplanes arrangement $\mathcal{B}_{m,n} = \{H_{ij} \mid i = 1, \ldots, m, j = 1, \ldots, n\}$ defined in (2.6). Then, $\mathcal{B}_{m,n}$ is stable by the permutation action, i.e., for any $\sigma \in S_n$, $\sigma(H_{ij}) = H_{i\sigma^{-1}(j)}$ holds, where $\sigma(H_{ij}) = \{\sigma \cdot \boldsymbol{x} \mid \boldsymbol{x} \in H_{ij}\}$. Then, the set of chambers $\mathrm{Ch}(\mathcal{B})$ is also stable by permutation action. We remark that the measure of complexity $c^{\sim}(\mathcal{H}_K^{\mathrm{inv}}(n, mn, m'))$ is equal to the maximum number of orbits of $\mathrm{Ch}(\mathcal{B})$, because by perturbing weight matrix or bias, we may assume that any two chambers in different orbits have different volumes. We set $\mathcal{A}_n$ to be the arrangement $\{W_{ij} \mid 1 \leq i < j \leq n\}$ called the *Coxeter arrangement* of S_n, where W_{ij} is the hyperplane defined by the equation $x_i - x_j = 0$. We may assume that $\mathcal{A}_n \cap \mathcal{B}_{m,n} = \emptyset$ by perturbing weight matrix or bias vector if required. Let $\mathcal{C}_{m,n} = \mathcal{A}_n \cup \mathcal{B}_{m,n}$. Then, by (Kamiya et al., 2012, Th. 2.6), the following holds:

Theorem 2. *The number of orbits of* $\mathrm{Ch}(\mathcal{B}_{m,n})$ *with respect to permutation action is equal to* $|\mathrm{Ch}(\mathcal{C}_{m,n})|/n!$.

This theorem allows us to reduce the calculation of the number of *orbits of chambers* of $\mathrm{Ch}(\mathcal{B}_{m,n})$ to the calculation of the number $|\mathrm{Ch}(\mathcal{C}_{m,n})|$ of chambers of $\mathrm{Ch}(\mathcal{C}_{m,n})$. This can be calculated inductively using the Deletion-Restriction theorem (Theorem 1 in Appendix B). Then, we obtain the following estimate of the complexity of permutation-invariant shallow model:

Theorem 3. *The measure of complexity of* $\mathcal{H}_K^{\mathrm{inv}}(m, mn, m')$ *satisfies* $c^{\sim}(\mathcal{H}_K^{\mathrm{inv}}(m, mn, m')) \leq (n + \alpha)!/\alpha! n!$. *Here,* $\alpha = 2^{mH(1/m)}$ *and* $\gamma!$ *for a positive real number* γ *is the generalized factorial defined by* $\gamma! = \prod_{0 \leq k < \gamma} (\gamma - k)$.

Proof. We set c_n^k as the numbers of the chambers of the hyperplane arrangement $\mathcal{C}_{m,n}^k = \mathcal{A}_k \cup \mathcal{B}_{m,n}$ for

$$\mathcal{A}_k = \{W_{ij} \mid 1 \leq i < j \leq k\}.$$

This $\mathcal{A}_k$ can be regarded as the Coxeter arrangement for S_k. Using this notation, it is straightforward to demonstrate that c_n^k satisfies the following recurrence relation:

$$c_n^k = c_n^{k-1} + k c_{n-1}^{k-1}.$$

Using this relation, we have

$$|\mathrm{Ch}(\mathcal{C}_{m,n})| = c_n^n = \sum_{l=0}^{n} \left(\sum_{1 \le k_1 < \cdots < k_l \le n} k_1 \cdots k_l \right) c_{n-l}^0.$$

If we use the upper bound of $c_{n-l}^0 \le \alpha^{n-l}$, where $\alpha = 2^{mH(1/m)}$ as in (2.4), we have

$$\begin{aligned} |\mathrm{Ch}(\mathcal{C}_{m,n})| &\le \sum_{l=0}^{n} \left(\sum_{1 \le k_1 < \cdots < k_l \le n} k_1 \cdots k_l \right) \alpha^{n-l} \\ &= \prod_{k=1}^{n} (\alpha + k) = \frac{(n+\alpha)!}{\alpha!}. \end{aligned}$$

Hence, by combining this and Theorem 2, the number of orbits of $\mathrm{Ch}(\mathcal{B}_{m,n})$ is bounded from above as

$$\begin{aligned} (\text{the number of orbits of } \mathrm{Ch}(\mathcal{B}_{m,n})) &= \frac{|\mathrm{Ch}(\mathcal{C}_{m,n})|}{n!} \\ &\le \frac{(n+\alpha)!}{\alpha! n!}. \end{aligned}$$ □

3.4 Comparison of the measures between fully connected and permutation invariant models

We compare these complexities between fully connected shallow model and permutation invariant shallow model. To equalize the number of hidden units in both models, we consider $n_0 = n$ and $n_1 = mn$. Let K be a connected compact n-dimensional subset $K \subset \mathbb{R}^n$ which is stable by permutation action. Then, because the maximum number of equivalent classes for fully connected shallow models is bounded from below by $\alpha^n / \sqrt{8n(1-1/m)}$ as in (2.4), where $\alpha = 2^{mH(1/m)}$. This means that the measure of complexity increases exponentially when n increases. Meanwhile, by Theorem 3, the maximum number of equivalent classes for permutation invariant shallow models is bounded from above by

$$\frac{(n+\alpha)!}{\alpha! n!} \le \frac{(n+\alpha)(n+\alpha-1)\cdots(n+\alpha-\lfloor\alpha\rfloor)}{\alpha!}.$$

In the second inequality, we used the fact that $n+\alpha-\lfloor\alpha\rfloor-k \le n-k+1$. By this argument, the measure of complexity $c^{\sim}(\mathcal{H}_K^{\mathrm{inv}}(m, mn, n'))$ of the set of the permutation invariant shallow models is bounded from above by a polynomial with respect to n of degree $\lfloor\alpha\rfloor + 1$. By comparing these measures, we have

$$\begin{aligned} c^{\sim}(\mathcal{H}_K^{\mathrm{inv}}(n, mn, n')) &\le \frac{(n+\alpha)!}{\alpha! n!} \\ &\ll \frac{\alpha^n}{\sqrt{8n(1-1/m)}} \\ &\le c^{\sim}(\mathcal{H}_K^{\mathrm{full}}(n, mn, n')). \end{aligned}$$

In particular, $c^{\sim}(\mathcal{H}_K^{\mathrm{inv}}(m, mn, n'))$ is strictly smaller than $c^{\sim}(\mathcal{H}_K^{\mathrm{full}}(m, mn, n'))$. Therefore, the proposed complexity behaves better to evaluate expressive power than simply counting linear regions.

4 Specific deeper models

In this section, we provide a variant of the model which has been introduced by Montúfar et al. (2014) and show that this can be used to confirm that deep models can have much higher complexity than shallow models.

4.1 A variant of the model of Montúfar et al

We here introduce a variant of the model of Montúfar et al. (2014). The original model introduced by Montúfar et al. (2014) is a deep neural network defined by some special affine maps designed to cause "folding" efficiently. From the way it is constructed, the hidden layers divide the input space into a grid of hypercubes, and the division into linear regions produced by the output layer is copied into each hypercube. We modify this model to be able to control the lengths of sides of the hypercubes to obtain hypercuboids which have different volumes.

The model is defined as follows: We consider a neural network of depth $L+1$ and width as (2.1). We assume that $n \le n_l$ for any l and set $p_l = \lfloor n_l/n \rfloor$. For $j \in \{1, 2, \ldots, n\}$, we set $\boldsymbol{w}_j^\top = (0, \ldots, 0, 1, 0, \ldots, 0)$ as the vector $\boldsymbol{w}_j \in \mathbb{R}^n$ whose j-th entry is 1 and the others are 0. For $l = 1, 2, \ldots, L-1$, we define $\widetilde{h}^{(l)} \colon \mathbb{R}^n \to \mathbb{R}^n$ as follows. We take, for any $j = 1, 2, \ldots, n$, positive integers $a_1^{j,(l)}, \ldots, a_{p_l}^{j,(l)}$ satisfying $\sum_{k=1}^{p_l} a_k^{j,(l)} = 1$ and set $b_k^{j,(l)} = (a_k^{j,(l)})^{-1}$ and

$$c_k^{j,(l)} = \begin{cases} -b_k^{j,(l)}(a_1^{j,(l)} + \cdots + a_k^{j,(l)}) & \text{if } k \text{ is even,} \\ -b_k^{j,(l)}(a_1^{j,(l)} + \cdots + a_{k-1}^{j,(l)}) & \text{if } k \text{ is odd.} \end{cases}$$

For $j = 1, 2, \ldots, n$ and $k = 1, \ldots, p_l$, we define the function $h_k^{j,(l)} \colon \mathbb{R}^n \to \mathbb{R}$ as

$$h_k^{j,(l)}(\boldsymbol{x}) = \begin{cases} \max\{0, b_1^{j,(l)} \boldsymbol{w}_j^\top \boldsymbol{x}\} & \text{if } k = 1, \\ \max\{0, (b_{k-1}^{j,(l)} + b_k^{j,(l)}) \boldsymbol{w}_j^\top \boldsymbol{x} + \sum_{s=2}^{k} c_s^{j,(l)}\} & \text{if } k \ge 2. \end{cases}$$

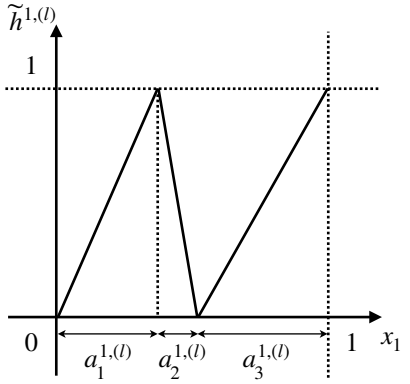

Figure 6: The graph of $\widetilde{h}^{1,(l)}$ for $p = 3$

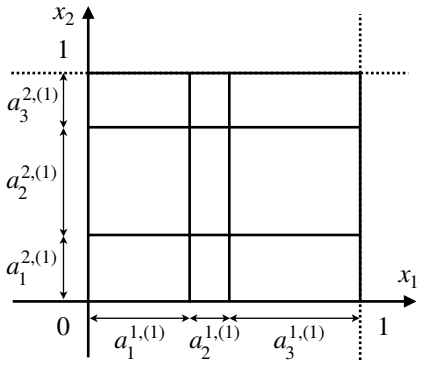

Figure 7: A decomposition of $[0,1]^2$ into rectangles (2-dim hypercuboids) by $\widetilde{h}^{(l)}$ for $n = 2, p = 3$

Using these $h_k^{j,(l)}$, we define the map $\widetilde{h}^{j,(l)} \colon \mathbb{R}^n \to \mathbb{R}$ as

$$\widetilde{h}^{j,(l)}(\boldsymbol{x}) = \sum_{k=1}^{p} (-1)^{k-1} h_k^{j,(l)}(\boldsymbol{x}).$$

Then, as in Figure 6, for $\boldsymbol{x} \in \mathbb{R}^n$ such that $a_1^{j,(l)} + \cdots + a_{i-1}^{j,(l)} \leq x_j < a_1^{j,(l)} + \cdots + a_i^{j,(l)}$, $\widetilde{h}^{j,(l)}(\boldsymbol{x})$ satisfies

$$\widetilde{h}^{j,(l)}(\boldsymbol{x}) = (-1)^{i+1}(b_i^{j,(l)} x_j + c_i^{j,(l)}).$$

We remark that although the input space of $\widetilde{h}^{j,(l)}$ is $\mathbb{R}^n$, this map depends only on j-th entry of $\boldsymbol{x}$. Hence, we can regard this as from $\mathbb{R}$ to $\mathbb{R}$. Moreover, this map $\widetilde{h}^{j,(l)}$ divides the subinterval $[0,1]$ of x_j-axis into p_l regions $(-\infty, 0]$, $[0, a_1^{j,(l)}]$, $[a_1^{j,(l)}, a_1^{j,(l)} + a_2^{j,(l)}]$, $\ldots$, $[\sum_{i=1}^{p_l-1} a_i^{j,(l)}, \infty)$ and the image of each regions by $\widetilde{h}^{j,(l)}$ is $[0,1]$. This construction makes a p_l-fold "folding".

We define $\widetilde{h}^{(l)} \colon \mathbb{R}^n \to \mathbb{R}^n$ by $\widetilde{h}^{(l)} = (\widetilde{h}^{1,(l)}, \ldots, \widetilde{h}^{n,(l)})^\top$. By the construction, this map $\widetilde{h}^{(l)}$ can be realized as a ReLU neural network as

$$\begin{aligned} &\mathbb{R}^n \to \mathbb{R}^{n_l} \to \mathbb{R}^n; \\ &\boldsymbol{x} \mapsto (h_1^{1,(l)}(\boldsymbol{x}), \ldots, h_{p_l}^{n,(l)}(\boldsymbol{x}), 0, \ldots, 0)^\top \\ &\mapsto (\widetilde{h}^{1,(l)}(\boldsymbol{x}), \ldots, \widetilde{h}^{n,(l)}(\boldsymbol{x}))^\top. \end{aligned}$$

This map $\widetilde{h}^{(l)}$ divides $[0,1]^n \subset \mathbb{R}^n$ into p_l^n n-dimensional hypercuboids. We remark that the volume of the $(i_1, \ldots, i_n)$-th hypercuboid is $a_{i_1}^{1,(l)} a_{i_2}^{2,(l)} \cdots a_{i_n}^{n,(l)}$ as in Figure 7.

Then, the composition $\widetilde{h}^{(L-1)} \circ \cdots \circ \widetilde{h}^{(1)}$ defines the deep neural network of depth L, width $n_0, n_1, \ldots, n_L$, and output $\mathbb{R}^n$. This map sends $[0,1]^n \in \mathbb{R}^n$ to $[0,1]^n \subset \mathbb{R}^n$ and divides $[0,1]^n$ into the $(p_1 p_2 \cdots p_{L-1})^n$ n-dimensional hypercubes with linear regions as in Figure 8. Then, the volume of $(\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n)$-th hypercube is

$$\begin{aligned} &\left(a_{i_{1,L-1}}^{1,(L-1)} \cdots a_{i_{11}}^{1,(1)}\right) \cdot \left(a_{i_{2,L-1}}^{2,(L-1)} \cdots a_{i_{21}}^{2,(1)}\right) \cdot \\ &\qquad \cdots \cdot \left(a_{i_{n,L-1}}^{n,(L-1)} \cdots a_{i_{n1}}^{n,(1)}\right), \end{aligned}$$

where $\bar{i}_k = (i_{k1}, \ldots, i_{k,L-1}) \in \prod_{l=1}^{L-1} \{1, \ldots, p_l\}$.

In particular, by perturbing weights if necessary, we may assume that any hypercuboids have different volumes.

Next, we choose a map $F \colon \mathbb{R}^n \to \mathbb{R}^{n_L}$ which gives a hyperplane arrangement whose chambers have different volumes introduced in Section 3.2 and by scaling, we assume that all the intersections of hyperplanes are in the interior of the hypercube $[0,1]^n \subset \mathbb{R}^n$. Finally, we take the composition $\widetilde{h}^{(L-1)} \circ \cdots \circ \widetilde{h}^{(1)}$ with F. Then, the hyperplanes arrangement in $[0,1]^n$ defined by F is copied into each hypercuboids as in Figure 8. If we need, by perturbing weights again, we may assume that any linear region has different volume. This implies that the measure of complexity $c^\sim(F \circ \widetilde{h}^{(L-1)} \circ \cdots \circ \widetilde{h}^{(1)})$ coincides with the maximum of the number of linear regions. In particular, this is equal to

$$\prod_{i=1}^{L-1} \left(\left\lfloor \frac{n_i}{n} \right\rfloor\right)^n \left(\sum_{k=0}^{n} \binom{n_L}{k}\right).$$

This shows the following:

Theorem 4. *The measure of complexity $c^\sim(\mathcal{H}^{\mathrm{full}}_{[0,1]^n}(n_0, n_1, \ldots, n_L, n_{L+1}))$ for the model of above defined ReLU deep neural networks is bounded from below by $\prod_{i=1}^{L-1} \left(\left\lfloor \frac{n_i}{n} \right\rfloor\right)^n \left(\sum_{k=0}^{n} \binom{n_L}{k}\right)$.*

As a consequence of the arguments of Section 3.2 and this section, both of the complexities for fully connected models which appear there are same as maximum numbers of linear regions. Hence, by similar argument to Montúfar et al. (2014), the complexity of deeper models is exponentially larger than the shallow models. This also shows the benefit of depth for neural network.

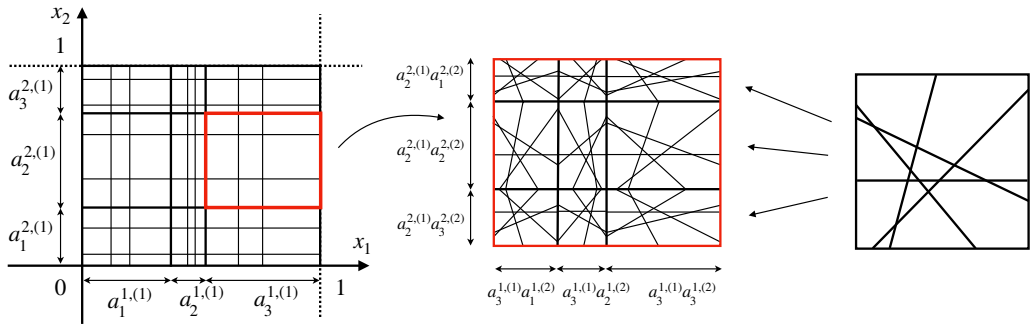

Figure 8: A grid decomposition of $[0,1]^2$ into rectangles by $\widetilde{h}^{(2)} \circ \widetilde{h}^{(1)}$ for $n = 2, p_1 = p_2 = 3$ and an image of copies of hyperplane arrangement in $[0,1]^2$ by F into the rectangles

4.2 A benefit of depth for deep set models

We here consider a permutation-invariant deep model, called *deep set model* introduced by Zaheer et al. (2017). This model is made by stacking some permutation equivariant maps and one invariant map. Thus, the obtained map is permutation-invariant. This model has some common features with the model of Montúfar et al. (2014). Indeed, the original model of Montúfar et al, except for the map from the last hidden layer to the output layer, is equivalent to the deep set model. We shall modify the model variant introduced in Section 4.1 to be a deep set model and show that deep set models also have a similar benefit of depth.

As mentioned above, the deep set model is defined by stacking permutation equivariant affine maps and one invariant map. More specifically, the ReLU deep neural network $f_{L+1} \circ \mathrm{ReLU} \circ f_L \circ \cdots \circ \mathrm{ReLU} \circ f_1$ for affine maps $f_i \colon (\mathbb{R}^n)^{m_{i-1}} \to (\mathbb{R}^n)^{m_i}$ is called *a deep set model* if $f_1, \ldots, f_L$ are permutation equivariant and $f_{L+1} \colon (\mathbb{R}^n)^{m_L} \to \mathbb{R}^{m_{L+1}}$ is permutation-invariant. For $\underline{m} = (m_1, m_2, \ldots, m_L, m_{L+1})$, let $\mathcal{H}^{\mathrm{inv}}_{[0,1]^n}(n, \underline{m})$ be the set of the restrictions to $[0,1]^n$ of the deep set models.

If we assume that the variant of model of Montúfar et al. (2014) which we introduced in Section 4.1 satisfies that $h_k^{1,(l)} = \cdots = h_k^{n,(l)}$ for any k and any l, and that $F \colon (\mathbb{R}^n)^{m_L} \to \mathbb{R}^{m_{L+1}}$ is permutation-invariant, then the obtained neural network $F \circ \widetilde{h}^{(L-1)} \circ \cdots \circ \widetilde{h}^{(1)}$ is in $\mathcal{H}^{\mathrm{inv}}_{[0,1]^n}(n, \underline{m})$. In this case, $a_k^{1,(l)} = \cdots = a_k^{n,(l)}$ holds for any k and l. We set $a_k^{(l)}$ to be this number. The obtained neural network providing the $(\prod_{i=1}^{L-1} p_l)^n$ n-dimensional hypercuboids. However, the volume of $(\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n)$-th hypercuboid is

$$\left(a^{(L-1)}_{i_{1,L-1}} \cdots a^{(1)}_{i_{11}}\right) \cdot \left(a^{(L-1)}_{i_{2,L-1}} \cdots a^{(1)}_{i_{21}}\right) \cdots \left(a^{(L-1)}_{i_{n,L-1}} \cdots a^{(1)}_{i_{n1}}\right),$$

where $\bar{i}_k = (i_{k,1}, \ldots, i_{k,L-1}) \in \prod_{l=1}^{L-1} \{1, \ldots, p_l\}$. We regard the index set $\prod_{l=1}^{L-1} \{1, \ldots, p_l\}$ as an ordered set by the lexicographic order $\leq$. Then, by perturbing the weights or biases if we need, we may assume that any hypercuboid in the set of hypercuboids whose index $(\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n)$ satisfies $\bar{i}_1 < \bar{i}_2 < \cdots < \bar{i}_n$ have different volumes, and the number of such hypercuboids is $\binom{p_1 \cdots p_{L-1}}{n}$. We choose the affine map from $\mathbb{R}^n$ to output layer $(\mathbb{R}^n)^{m_L}$ to be the one which achieves the measure of complexity of $\mathcal{H}^{\mathrm{inv}}_{[0,1]^n}(n, nm_L, m_{L+1})$ as in Section 3.3. Hence, the measure of complexity of $\mathcal{H}^{\mathrm{inv}}(n, \underline{m})$ is bounded from below by $C \cdot (m_1 \cdots m_L)^n n^n / n!$ for a positive constant C. In particular, the following holds:

Theorem 5. *The following holds:* $c^{\sim}(\mathcal{H}^{\mathrm{inv}}_{[0,1]^n}(n, \underline{m})) = \Omega((m_1 \cdots m_L)^n (n^n/n!)) = \Omega((m_1 \cdots m_L e)^n / \sqrt{n})$.

We compare this with the shallow invariant model having same number of hidden units. Then, the width of the hidden layer of the shallow model is equal to $n \sum_{i=1}^{L} m_i$. By the argument in Section 2.2, the measure of complexity is $\Theta((\sum_{i=1}^{L} m_i)^n \cdot e^n / \sqrt{n})$. This yields that for the deep set model, deeper models can obtain exponentially more complexity than shallow models in our measure.

5 Conclusion

In this paper, we defined a new measure of complexity of ReLU neural networks, which is closer to expressive power than the number of linear regions. Specifically, we considered fully connected and Permutation-invariant models as examples, which are indistinguishable from the conventional measure of linear regions but have different expressive power. The new complexity is introduced as the number of equivalence classes that identify linear regions and linear functions on them with those transferred by a Euclidean transformation. Considering that, we have shown that the values of the measure for the two networks above are actually different. In this sense, the proposed measure of complexity can be considered to represent the expressive power of the function more closely. We also proved that the value of the proposed measure increases exponentially for deeper networks by refining the model of Montúfar et al. (2014) for both the fully connected model and the deep set model.

Acknowledgments

The authors would like to thank the anonymous reviewers for their suggestions and helpful comments. This work was supported in part by the Grant for Basic Science Research Projects from The Sumitomo Foundation (No.200484) and the JSPS Grant-in-Aid for Scientific Research C (20K03743).

References

Alex, K., Sutskever, Ilya, S., and Geoffrey E, H. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. (2016). Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*.

Ash, R. (1965). *Information theory*. Interscience Tracts in Pure and Applied Mathematics, No. 19. Interscience Publishers John Wiley & Sons, New York-London-Sydney.

Bianchini, M. and Scarselli, F. (2014). On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems*, 25(8):1553–1565.

Chatziafratis, V., Nagarajan, S. G., and Panageas, I. (2020). Better depth-width trade-offs for neural networks through the lens of dynamical systems. In *International Conference on Machine Learning*, pages 1469–1478. PMLR.

Chatziafratis, V., Nagarajan, S. G., Panageas, I., and Wang, X. (2019). Depth-width trade-offs for relu networks via sharkovsky's theorem. In *International Conference on Learning Representations*.

Eldan, R. and Shamir, O. (2016). The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. (2013). Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*.

Hanin, B. and Rolnick, D. (2019). Complexity of linear regions in deep networks. *arXiv preprint arXiv:1901.09021*.

Kamiya, H., Takemura, A., and Terao, H. (2012). Arrangements stable under the coxeter groups. In *Configuration spaces*, pages 327–354. Springer.

Maron, H., Fetaya, E., Segol, N., and Lipman, Y. (2019). On the universality of invariant networks. *Proceedings of the 36th International Conference on Machine Learning*, 97.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932.

Orlik, P. and Terao, H. (2013). *Arrangements of hyperplanes*, volume 300. Springer Science & Business Media.

Pascanu, R., Montúfar, G., and Bengio, Y. (2013). On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Dickstein, J. S. (2017). On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2847–2854. JMLR. org.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pages 4558–4566.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

Sonoda, S. and Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268.

Telgarsky, M. J. (2016). Benefits of depth in neural networks. *Journal of Machine Learning Research*, 49(June):1517–1539.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In *Advances in neural information processing systems*, pages 3391–3401.